

03-13-20

ASSISTANT COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, DC 20231

PATENT  
Date: March 10, 2000  
File No. 0828.63692



Sir:

Transmitted herewith for filing is the patent application of

Inventor(s): Toyoaki Furusawa, Haruhito Nagata,  
Toshihiro Nishimura

I hereby certify that this paper is being deposited with the United States Postal Service as Express Mail in an envelope containing the above application, addressed to: Asst. Comm. for Patents, Washington, D.C. 20231, on the date.

03/10/00  
Date

*S. G.*  
Express Mail Label No.: EL409491149US

For: MESSAGE PROCESSING APPARATUS

Enclosed are:

- (X) 36 pages of specification, including 7 claims and an abstract.  
 (X) an executed oath or declaration, with power of attorney.  
 ( ) an unexecuted oath or declaration, with power of attorney.  
 ( )        sheet(s) of informal drawing(s).  
 (X) 24 sheet(s) of formal drawings(s).  
 (X) Assignment(s) of the invention to FUJITSU LIMITED.  
 (X) Assignment Form Cover Sheet.  
 (X) A check in the amount of \$40.00 to cover the fee for recording the assignment(s) is enclosed.  
 (X) Information Disclosure Statement.  
 (X) Form PTO-1449 and cited references.  
 ( ) Associate power of attorney.  
 (X) Priority Document.

Fee Calculation For Claims As Filed

a)	Basic Fee		\$ 690.00
b)	Independent Claims	<u>2</u> - <u>3</u> = <u>0</u>	x \$ 78.00 = \$_____
c)	Total Claims	<u>7</u> - <u>20</u> = <u>0</u>	x \$ 18.00 = \$_____
d)	Fee for Multiple Claims		\$260.00 = \$_____
Total Filing Fee			<u>\$ 690.00</u>

- ( )        Statement(s) of Status as Small Entity, reducing Filing Fee by half to \$\_\_\_\_\_
- (X) A check in the amount of \$690.00 to cover the filing fee is enclosed.
- (X) The Commissioner is hereby authorized to charge any additional fees which may be required to this application under 37 C.F.R. §§1.16-1.17, or credit any overpayment, to Deposit Account No. 07-2069. Should no proper payment be enclosed herewith, as by a check being in the wrong amount, unsigned, post-dated, otherwise improper or informal or even entirely missing, the Commissioner is authorized to charge the unpaid amount to Deposit Account No. 07-2069. A duplicate copy of this sheet is enclosed.

Suite 8660 - Sears Tower  
233 S. Wacker Drive  
Chicago, Illinois 60606  
(312) 993-0080

GREER, BURNS & CRAIN, LTD.

*Patrick G. Burns*  
By: Patrick G. Burns  
Patrick G. Burns  
Registration No. 29,367

MESSAGE PROCESSING APPARATUS

BACKGROUND OF THE INVENTION

1. Field of the Invention

5       The present invention relates to a message processing apparatus, and more particularly, to a message processing apparatus which receives and displays messages sent from sending stations.

2. Description of the Related Art

10       The proliferation of personal computers has been promoting message communications over electronic media, including electronic mail exchange and net news reading. As people receive more and more messages every day, there will be an urgent need to efficiently deal with a large  
15       number of incoming messages, sorting out important ones from others.

      To meet the above need, some conventional message processing apparatus handling e-mail and other electronic messages provide a message flag function which allows the  
20       sender to set a certain flag when sending what he/she thinks is an important message. At the receiving end, messages appearing on a display screen are marked with a special symbol (e.g., exclamation mark "!" ) if their flags are set. Some other conventional apparatus offer e-mail  
25       filters which put the received messages into separate folders or make some of them invisible to the recipient, effectively screening them by sender names or some

I hereby certify that this paper is being deposited with the United States Postal Service as Express Mail in an envelope addressed to: Asst. Comm. for Patents, Washington, D.C. 20231, on this date.

-1-

03-10-00  
Date

Express Mail Label No.: EL409491149US

keywords contained in the messages. The mail recipients can set up such filters to meet their individual needs.

The above message screening functions, however, are limited in their capabilities. Particularly, they 5 provide only simple visual distinctions between important messages and unimportant ones, which are so insufficient that the recipient sometimes fails to catch important notes or even discards them.

Another deficiency is non-uniformity in the 10 behavior of receiving ends. In conventional message processing systems, the individual users have to set up their systems by themselves. Thus the receiving systems may react to the same message in different ways depending on their individual setups.

15

#### SUMMARY OF THE INVENTION

Taking the above into consideration, an object of the present invention is to provide a message processing apparatus which processes received messages according to 20 their content, in such a way that many users share the same processing methods.

To accomplish the above object, according to the present invention, there is provided a message processing apparatus for receiving and displaying messages sent from 25 a sending end. This message processing apparatus comprises a receiver, a key information extraction unit, a program loading unit, and an execution unit. The receiver receives

a message sent from the sending end. The key information extraction unit extracts key information from the message received by the receiver. The key information is contained in a message as part of its attribute data, and used to  
5 identify a predetermined handler program relevant to the message. The program loading unit loads the predetermined handler program associated with the extracted key information. The execution unit executes the handler program loaded by the loading unit at prescribed times.

10 The above and other objects, features and advantages of the present invention will become apparent from the following description when taken in conjunction with the accompanying drawings which illustrate preferred embodiments of the present invention by way of example.

15

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a conceptual view of the present invention;

20 FIG. 2 is a diagram which shows a typical embodiment of the present invention;

FIG. 3 is a diagram which shows the details of a distributing station shown in FIG. 2;

FIG. 4 is a diagram which shows the details of a receiving station shown in FIG. 2;

25 FIG. 5 is a flowchart which explains a typical process executed by a receiving station when downloading a keyword lookup table and primitive process description

from the distributing station;

FIG. 6 is a diagram which shows a typical keyword lookup table;

FIG. 7 is a diagram which shows a typical  
5 primitive process description;

FIG. 8 is a diagram which shows a typical screen  
of a mailer application program designed to receive e-mail  
messages;

FIG. 9 is a flowchart showing a typical process to  
10 be executed to identify a specific plug-in that is  
relevant to a received e-mail message;

FIG. 10 is a diagram showing an example of a  
received e-mail message;

FIG. 11 is a flowchart showing a typical process  
15 to be executed when an e-mail message with a keyword  
"<URGENT: BY \*\*\*\*>" is received;

FIG. 12 is a diagram which shows a typical screen  
displayed on the monitor unit as a result of the process  
of FIG. 11;

20 FIG. 13 is a flowchart showing a typical process  
to be executed when the mailer application program is  
launched again after being closed without opening the e-  
mail message with a keyword "<URGENT: BY \*\*\*\*>,"

FIG. 14 is a diagram showing a typical screen  
25 which appears on the monitor unit as a result of the  
process of FIG. 13;

FIG. 15 is a flowchart showing a typical process

to be executed when the user attempts to delete an e-mail message containing a keyword "<URGENT: BY \*\*\*\*>" without opening it;

FIG. 16 is a diagram showing a typical screen that appears on the monitor unit as a result of the process of FIG. 15;

FIG. 17 is a flowchart showing a typical process to be executed when an e-mail message containing a keyword "<MUST-READ>" is received;

FIG. 18 is a diagram showing a typical screen that appears on the monitor unit as a result of the process of FIG. 17;

FIG. 19 is a flowchart showing a typical process to be executed when the mailer application program is launched again after being closed without opening a file attachment that is delivered in an e-mail message with a keyword "<MUST-READ>;"

FIG. 20 is a diagram showing a typical screen that appears on the monitor unit as a result of the process of FIG. 19;

FIG. 21 is a flowchart showing a typical process to be executed when the user attempts to delete an e-mail message containing a keyword "<MUST-READ>;"

FIG. 22 is a diagram showing a typical screen that appears on the monitor unit as a result of the process of FIG. 21;

FIG. 23 is a flowchart showing a typical process

to be executed when an e-mail message containing a keyword "<TICKER>" is received; and

FIG. 24 is a diagram showing a typical screen that appears on the monitor unit as a result of the process of

5 FIG. 23.

#### DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred embodiments of the present invention will be described below with reference to the accompanying drawings.

FIG. 1 is a conceptual view of the present invention, illustrating a local area network (LAN) environment which is linked to another network 1 through a gateway 2. The network 1 is a wider networking environment such as the Internet, where data packets are transmitted back and forth. The gateway 2 enables a distributing station 4 and receiving station 5 on the local area network to gain access to other devices on the network 1. For this purpose, the gateway 2 provides various services, including message format transformation, address translation, and protocol conversion. Network cables 3 provide LAN interconnections between the distributing station 4, receiving stations 5, and other network nodes not shown in FIG. 1. While FIG. 1 illustrates only one receiving station 5, it has to be appreciated that there are a plurality of like stations in a real-world system.

The distributing station 4 stores handler program

100-100000000000

files 4a and a table 4b. The handler program files 4a provide a collection of processes which the receiving station 5 will execute when it receives a message containing particular key information, or keywords. The 5 table 4b defines the relationships between such keywords and their associated handler programs. The system administrator or other authoritative persons maintain the contents of those handler program files 4a and table 4b. When they are updated, the distributing station 4 sends a 10 request message to apply the update to corresponding local files in the receiving station 5.

The receiving station 5 comprises a receiver 5a, a key information extraction unit 5b, a program loading unit 5c, an execution unit 5d, a downloading unit 5e, handler 15 program files 5f, and a table 5g. With those functional elements, the receiving station 5 receives and displays messages sent from other stations. More specifically, the receiver 5a receives messages sent from a sending end. From each message received by the receiver 5a, the key 20 information extraction unit 5b extracts a keyword, or key information, to identify a particular handler program that is relevant to the message. Messages optionally contain such a keyword as part of their attribute data.

The program loading unit 5c installs a handler 25 program corresponding to a given keyword. Here, the term "loading" refers to an operation of transferring a program from the computer's secondary storage to main memory,

thereby making it ready to execute. The execution unit 5d executes the handler program prepared by the program loading unit 5c at appropriate times.

The downloading unit 5e downloads the handler  
5 program files 4a and the table 4b to update its local  
handler program files 5f and table 5g when it is requested  
by the distributing station 4. The handler program files  
5f are a replica of the files 4a stored in the  
distributing station 4, providing a collection of  
10 processes associated with each keyword. The table 5g is a  
replica of the table 4b in the distributing station 4,  
which defines the relationships between message keywords  
and their associated handler programs.

The above system of FIG. 1 will operate as follows.  
15 Suppose, for example, that a certain station (not shown)  
on the network has created and transmitted to the  
receiving station 5 a message containing a keyword "<MUST-  
READ>" in its subject line, one of the message's attribute  
data fields. This message is received by the receiver 5a  
20 in the receiving station 5 and supplied to the key  
information extraction unit 5b. The key information  
extraction unit 5b takes out a keyword from the message  
and supplies it to the program loading unit 5c. In the  
present example, the keyword "<MUST-READ>" is extracted  
25 and supplied to the program loading unit 5c.

The program loading unit 5c first refers to the  
table 5g to determine which handler program is relevant to

the extracted keyword "<MUST-READ>." The program loading unit 5c then loads the selected handler program to the system, converting it into executable form. The loaded handler program provides a collection of primitive processes. Now that the program is loaded, the execution unit 5d is ready to choose and execute an appropriate primitive process in response to an event that may occur to the received message. The handler program will be removed when its corresponding message is deleted, because 10 it is loaded in association with each individual message.

With the handler program for the keyword "<MUST-READ>" loaded, the receiving station 5 first reacts to the event of "message reception" by displaying the following note on a monitor screen (not shown).

15 -- A must-read message has just arrived --  
Although this note prompts the recipient to read the message, the user may or may not open it immediately. If he/she reads the message at once, no further primitive processes will be executed, and the system removes the 20 handler program when the user deletes the message. The recipient, however, may leave the message unread. If this is the case, the system will invoke a primitive process for another event "restart," the next time the user launches his/her mailer application. The system then 25 outputs another notification message as follows.

-- There is one must-read message in your inbox --

While this notification prompts again the recipient to

read the important message, he/she may even attempt to delete it by mistake. Or the mailer application may be programmed to erase the message when its storage time limit has expired. In such cases, the presence of keyword

5 "<MUST-READ>" invokes a primitive process corresponding to the "deletion" event, causing the system to display the following warning message, for example.

-- You are deleting a must-read message! --

As previously noted, the distributing station 4  
10 notifies the receiving stations 5 when any modification is made to the handler program files 4a or table 4b. In that case, the downloading unit 5e downloads new information from the distributing station 4 to update its local handler program files 5f and table 5g.

15 Thus, to summarize the above processing, the receiving station 5 receives a message, loads a handler program corresponding to key information (keyword) extracted from the received message, and executes appropriate primitive processes depending on the  
20 subsequent events. This mechanism keeps providing the user with timely notification about the received e-mail messages throughout their lifetime, from reception to deletion.

While FIG. 1 shows only one receiving station, two  
25 or more such stations can be connected to the network. Those receiving stations share the same set of handler program files and tables, since they are delivered from a

100-100000

single source, i.e., the distributing station 4. This naturally results in a unified behavior of the receiving stations when they receive a message containing a particular keyword. Further, the contents of the handler 5 program files 4a and table 4b can be centrally managed by the administrator of the distributing station 4. This liberates the users of the receiving stations 5 from burdensome tasks of keeping their local files up to date.

Referring now to FIGS. 2, 3, and so on, a 10 preferred embodiment of the present invention will be described in detail below. FIG. 2 shows a LAN environment which is linked to another network 10 via a gateway 20. The LAN interconnects a distributing station 40 and receiving stations 50 and 60. The network 10 is a wider 15 networking environment such as the Internet, where data packets are transmitted back and forth. The gateway 20 is a workstation-based network element, which provides various services including message format transformation, address translation, and protocol conversion to enable 20 data exchange between the LAN and the other network 10.

The distributing station 40 has a keyword lookup table that defines the relationships between message keywords and their associated handler programs. In the table, each handler program is referred to by its class name. The distributing station 40 also has primitive process descriptions that define a plurality of classes. 25 Each handler program provides a collection of processes

which the receiving stations 50 and 60 will execute when it receives a message containing particular key information, or keywords. The distributing station 40 requests the receiving stations 50 and 60 to update their local files, if the keyword lookup tables and/or primitive process descriptions in the distributing station 40 have been updated, or when the system administrator has instructed so.

FIG. 3 shows a detailed hardware configuration of the distributing station 40. As seen from FIG. 3, the distributing station 40 comprises a central processing unit (CPU) 40a, a read-only memory (ROM) 40b, a random access memory (RAM) 40c, a hard disk drive (HDD) 40d, and a graphics card 40e, an interface 40f, and a bus 40g. Additionally, an external monitor unit 45 is coupled to the distributing station 40.

The CPU 40a performs various operations according to the programs loaded on the RAM 40c, etc., besides controlling other parts of the distributing station 40. The ROM 40b stores basic system programs and data that the CPU 40a executes and manipulates. The RAM 40c serves as temporary storage for programs and data that the CPU 40a executes and manipulates at runtime. The HDD 40d stores the aforementioned keyword lookup table and primitive process descriptions, in addition to various application programs which the CPU 40a executes.

The graphics card 40e produces image data in

accordance with drawing commands from the CPU 40a and provides the monitor unit 45 with the resultant images in the form of video signals. The interface 40f is where a LAN cable 30 is terminated. It supports certain protocols  
5 to allow the distributing station 40 to communicate with the receiving stations 50 and 60. The monitor unit 45 is a cathode ray tube (CRT) monitor, for example, which displays text and graphic images according to the video signals supplied from the graphics card 40e.

10 Referring back to FIG. 2, the receiving stations 50 and 60 receive and display messages sent from other network nodes. They also provide various services according to the primitive processes defined in a specific primitive process description supplied from the  
15 distributing station 40.

FIG. 4 shows the details of the receiving station 50. Since the receiving station 50 and the receiving station 60 are similar in structure, the following explanation will focus on the receiving station 50. As  
20 seen from FIG. 4, the receiving station 50 comprises a CPU 50a, a ROM 50b, a RAM 50c, an HDD 50d, a graphics card 50e, an interface 50f, and a bus 50g. Additionally, some external input devices 51 and a monitor unit 55 are connected to the receiving station 50. Such elements of  
25 the receiving station 50 are similar to those that constitute the distributing station 40 explained in FIG. 3. Thus the details of their operation will not be described

here again.

The input devices 51 may include a keyboard and a mouse, for example, with which the user enters operating commands to the receiving station 50. The monitor unit 55 5 is a CRT monitor or other display device, as is the aforementioned monitor unit 45. It displays text and graphic images according to the video signals supplied from the receiving station 50.

The operation of the above system will now be 10 explained below, starting with a process to download a keyword lookup table and primitive process description from the distributing station 40 to the receiving station 50.

FIG. 5 is a flowchart which describes a typical 15 process to be executed by the receiving station 50 when downloading a keyword lookup table and primitive process description from the distributing station 40. The receiving station 50 calls this process in response to a request from the distributing station 40. Such requests 20 occur when some update is made to the keyword lookup table or primitive process descriptions in the distributing station 40, or when the administrator sends a certain instruction to the distributing station 40. When invoked, the process executes the following steps.

25 (S10) The CPU 50a determines whether there is any data update request from the distributing station 40. If there is, the process advances to step S11. If not,

the process is terminated.

(S11) The CPU 50a receives a keyword lookup table from the distributing station 40.

5 (S12) The CPU 50a receives primitive process descriptions from the distributing station 40.

(S13) The CPU 50a updates the keyword lookup table stored in the HDD 50d.

(S14) The CPU 50a updates the primitive process descriptions stored in the HDD 50d.

10 FIG. 6 shows an example of the keyword lookup table. This keyword lookup table provides four kinds of data items: type, primary sub-keyword, secondary sub-keyword, and associated plug-in. The first entry of the table, for instance, shows that the messages of type  
15 "MAIL" containing a primary sub-keyword "<URGENT:" and a secondary sub-keyword "BY \*\*\*\*>" (\*\*\*\* is a date code) are to be handled by its associated plug-in named "TWPURgentPlugin." As such, incoming messages may optionally contain a keyword in their subject line (or  
20 title) as one of the message attributes. Each keyword is generally composed of two substrings: primary sub-keyword and secondary sub-keyword. For example, "<URGENT: BY 99/06/14>" is a typical composite keyword inserted in the title of a message.

25 Every keyword is associated with a specific plug-in, which is a description (e.g., Java class) that defines a collection of primitive processes. FIG. 7 shows a

typical plug-in, named "TWPURgentPlugin," which appears in the keyword lookup table of FIG. 6. This plug-in consists of the following five primitive processes: initialization, message reception, application restart, message selection,  
5 and message deletion.

When a message is received, the receiving station 50 will operate as follows. Suppose here that the user of the receiving station 50 has operated the input devices 51 to run a certain application program to send and/or  
10 receive e-mail messages. FIG. 8 shows a window 70 to be displayed on the monitor unit 55 when such a mailer application program is started. On this example screen, the window 70 titled "E-MAIL" comes with a Minimize button 70a, a Maximize button 70b, and a Close button 70c aligned  
15 in the upper right corner for controlling the window 70. The window 70 further provides the following buttons for manipulating messages and controlling server connection. A Write button 70d is used when writing a new message. A Send/Receive button 70e is used when sending and/or  
20 receiving messages. The user presses a Connect button 70f or Disconnect button 70g when connecting or disconnecting his/her station to/from its local mail server. The pane 70h shows the current contents of the user's inbox. In this example of FIG. 8, the inbox stores two e-mail  
25 messages that have been opened by the user.

Suppose here that the user has pressed the Send/Receive button 70e. In response to that, the

receiving station 50 connects itself to the mail server and checks whether the server has any e-mail messages addressed to the user. If there are messages, the receiving station 50 downloads them and executes a process  
5 shown in the flowchart of FIG. 9. More specifically, the following steps are executed.

- (S30) The CPU 50a extracts a keyword, if any, from the subject line of the received e-mail message. Consider, for example, that the inbox holds an e-mail message  
10 shown in FIG. 10. The subject line of this message contains a keyword that reads "<URGENT: BY 99/06/14>," and the CPU 50a now extracts a primary sub-keyword "<URGENT:>" and a secondary sub-keyword "BY 99/06/14."
- 15 (S31) The CPU 50a determines whether the keyword has any arguments. If it has, the process advances to step S32. If not, the process proceeds to step S33.
- (S32) The CPU 50a extracts arguments from the keyword string. In the example of FIG. 10, an argument  
20 "99/06/14" is extracted.
- (S33) Consulting the keyword lookup table, the CPU 50a identifies which plug-in is associated with the keyword. In the present case, the composite keyword is "<URGENT: BY \*\*\*\*>," the CPU 50a obtains its relevant plug-in "TWPURgentPlugin" by referring to the keyword lookup table of FIG. 6.
- (S34) The CPU 50a makes access to the HDD 50d to read

out and load the plug-in that has been identified at step S33. In the present example, the CPU 50a reads a Java class "TWPURgentPlugin" out of the HDD 50d and creates an object corresponding to the plug-in. This  
5 object includes a collection of primitive processes and is linked to the e-mail message of interest. Accordingly, a relevant primitive process will be selected and invoked when a prescribed event occurs to that message. When the e-mail message of interest  
10 is deleted, its associated object will also be deleted at the same time. The next paragraph will focus on the primitive processes.

FIG. 11 is a flowchart showing the message reception process (FIG. 7), one of the primitive processes  
15 of the plug-in "TWPURgentPlugin." This process comprises the following steps.

(S40) The CPU 50a causes the monitor unit 55 to display a notification dialog box. FIG. 12 shows a typical screen which appears when a e-mail message of FIG. 10  
20 is received. Compared with FIG. 8, the pane 70h of the application program window 70 now shows the title of a newly arrived message, "Correction of Document." In addition, a notification dialog box 80 titled "URGENT" pops up over the application program window 70. The dialog box 80 has a Close button 80a in the upper right corner for closing the dialog box itself.  
25 The dialog box 80 shows the following notification on

its pane 80b.

-- Message "Correction of Document" has arrived.

This message calls for your action by 99/06/14. --

5 Further, at the bottom of the pane 80b, an OK button 80c is located to allow the recipient to acknowledge the notification.

(S41) The CPU 50a sets the Unread flag to the ON state to indicate that the recipient has not yet opened the e-mail message.

10 (S42) The CPU 50a determines whether the recipient has already opened the e-mail message in question. If he/she has, the process advances to step S43. If not, the process is terminated.

15 (S43) The CPU 50a shows the message text. In the present case, the monitor unit 55 presents a message window shown in FIG. 10.

(S44) The CPU 50a resets the Unread flag to the OFF state.

20 Through the above process steps, the receiving station 50 finds a keyword "URGENT" in an incoming e-mail message and produces a notification dialog box to inform the recipient that the received message requires timely action. This notification prompts the recipient to read the message and respond to it accordingly. The recipient, 25 however, may not open the messages immediately, in spite of the notification shown in FIG. 12. Referring now to FIG. 13, the next section will describe how the proposed system

100-10000000

will operate in such a situation.

Suppose that the recipient closes the application program without opening the newly arrived e-mail message shown in FIG. 12. Here, the recipient may even turn off 5 the power of the receiving station 50. The next time he/she launches the application program, the receiving station 50 executes the following process steps.

(S50) The CPU 50a determines whether the Unread flag of 10 the e-mail message in question is ON. If it is, the process advances to step S51. If not, the process is terminated. In the present example, the process advances to step S51 because the Unread flag was set to ON at step S41 in the process of FIG. 11 and the recipient closed the window of FIG. 12 without 15 resetting the flag by opening the message.

(S51) The CPU 50a displays a notification dialog box. FIG. 14 shows a typical layout of the notification dialog box 90 is titled "URGENT" and has a Close button 90a in the upper right corner for 20 closing the dialog box 90. The pane 90b shows a note telling that the recipient is requested to take action by June 14, 1999. At the bottom of the pane 90b, an OK button 90c is located to allow the recipient to acknowledge the notification.

25 (S52) The CPU 50a determines whether the recipient has opened the e-mail message in question. If so, the process advances to step S53. If not, the process is

terminated.

(S53) The CPU 50a causes the monitor unit 55 to display the message text.

(S54) The CPU 50a resets the Unread flag of the message 5 to the OFF state.

Through the above process steps, the restarted application program displays another notification dialog box to prompt the recipient to open the received message, even if he/she did not do so when the message arrived at 10 his/her inbox. This notification reminds the recipient of the unopened e-mail message that needs prompt action.

The receiving station 50 further generates a warning message, when the user attempts to delete an urgent e-mail message without opening it in spite of the 15 notification shown in FIG. 12. The receiving station 50 will also generate a like warning message when the mailer application program attempts to purge an e-mail message because of the expiration of its storage time limit that the inbox permits. Referring now to a flowchart of FIG. 15, 20 the following will describe the process executed in such situations. This process comprises the following steps.

(S60) The CPU 50a determines whether the Unread flag of 25 the e-mail message in question is in the ON state. If it is ON, the process advances to step S62. If not, the process proceeds to step S61. For instance, the process advances to step S62 when the recipient attempts to delete an unopened e-mail message

appearing on the screen of FIG. 12. To the contrary, the process goes to step S61 if the recipient has already opened the message.

5 (S61) The CPU 50a executes a routine to delete the specified e-mail message.

(S62) The CPU 50a displays a warning dialog box. FIG. 16 shows a typical warning dialog box 100, which is titled "URGENT" and has a Close button 100a in the upper right corner for closing the dialog box 100.  
10 The pane 100b now shows the following warning message.  
-- Message "Correction of Document" calls for your action by 99/06/14. Open and read it right now. -- At the bottom of the pane 100b, an OK button 100c is located to allow the recipient to acknowledge the  
15 above warning message.

(S63) The CPU 50a causes the monitor unit 55 to show the message text of interest.

(S64) The CPU 50a resets the Unread flag of the message to the OFF state.

20 Through the above process steps, the attempt to delete an unopened e-mail message causes a warning dialog box, as well as showing the message text. This prevents the recipient from deleting important e-mail messages accidentally.

25 The next section will focus on the primitive processes of another plug-in "MustReadPlugin" shown in FIG. 6. This plug-in handles e-mail messages with a must-see

5

file attachment. In short, the sender inserts a composite keyword "<MUST-READ>" to the subject line of a message to make sure that the recipient opens the message and its attachment. When such a message is received, the receiving station loads and executes the "MustReadPlugin."

Referring first to a flowchart of FIG. 17, the following will describe a process to be executed when an e-mail message with a must-see attachment is received. The process comprises several steps as follows.

10 (S70) The CPU 50a causes the monitor unit 55 to display a notification dialog box. FIG. 18 shows a typical layout of the notification dialog box. This dialog box 110 is titled "MUST-SEE FILE ATTACHMENT" and has a Close button 110a in the upper right corner for closing the dialog box 110. The pane 110b shows a note that reads as follows.

-- Message "Modification Required" has a must-see file attachment --

20 This note is followed by the name of the file attachment, "map.gif." Also, at the bottom of the pane 110b, an OK button 110c is placed to allow the recipient to acknowledge the above note.

(S71) The CPU 50a sets the Unopened flag to the ON state, thereby indicating that the recipient has not opened 25 the file attachment.

(S72) The CPU 50a determines whether the recipient has opened the file attachment in question. If so, the

process advances to step S73. If not, the process is terminated.

(S73) The CPU 50a causes the monitor unit 55 to open and show the file attachment.

5 (S74) The CPU 50a resets the Unopened flag to the OFF state.

Through the above process steps, the receiving station 50 produces a dialog box to notify the recipient of the arrival of an important file attachment, when it receives an e-mail message with a keyword "<MUST-READ>" in its subject line and is accompanied by a file attachment. This feature prevents the recipient from leaving the attachment unopened, while only reading the message text.

Suppose next that the recipient closes the application program without opening the received file attachment. He/she may even turn off the power of the receiving station 50. FIG. 19 is a flowchart showing a process to be executed when the application program is launched again in such a situation. This process comprises 20 the following steps.

(S80) The CPU 50a determines whether the Unopened flag of e-mail message in question is ON. If the flag is ON, the process advances to step S81. If not, the process is terminated.

25 (S81) The CPU 50a displays a notification dialog box on the monitor unit 55. FIG. 20 shows a typical notification dialog box 120, which is titled "MUST-

SEE FILE ATTACHMENT" and has a Close button 120a in the upper right corner for closing the dialog box 120. The pane 120b shows a note that reads as follows.

-- Message "Request for Modification" has a must-see  
5 file attachment --

Further, this note is followed by the name of the file attachment, "map.gif." Also, at the bottom of the pane 120b, an OK button 120c is placed to allow the recipient to acknowledge the above note.

10 (S82) The CPU 50a determines whether the recipient commands the receiving station 50 to open and show the file attachment of to the e-mail message in question. If so, the process advances to step S83. If not, the process is terminated.

15 (S83) The CPU 50a opens and shows the file attachment on the monitor unit 55.

(S84) The CPU 50a resets the Unopened flag of the message to OFF.

Through the above process steps, the receiving  
20 station 50 displays a notification dialog box when the application program is launched again, even if the recipient did not do so when the message arrived at his/her inbox. The notification reminds the recipient of the important file attachment that has not been opened.

25 The receiving station 50 further generates a warning message if the user attempts to delete an important e-mail message without opening its file

attachment. The receiving station 50 will also generate a like warning message when the mailer application program attempts to purge an e-mail message because of the expiration of its storage time limit that the inbox permits. Referring now to a flowchart of FIG. 21, the following will discuss what process will be executed in such situations. The process comprises the following steps.

5 (S90) The CPU 50a determines whether the Unopened flag of the e-mail message in question is ON. If it is, 10 the process advances to step S92. If not, the process proceeds to step S91.

(S91) The CPU 50a executes a routine to delete the specified e-mail message.

15 (S92) The CPU 50a displays a warning dialog box. FIG. 22 shows a typical layout of the warning dialog box. This dialog box 130 is titled "MUST-SEE FILE ATTACHMENT" and has a Close button 130a in the upper right corner for closing the dialog box 130. The pane 130b shows a message notifying the recipient of the 20 presence of a file attachment, as well as suggesting that the attachment is worth opening. An OK button 130c is located at the bottom of the pane 130b to allow the recipient to acknowledge the notification.

(S93) The CPU 50a opens and shows the file attachment on 25 the monitor unit 55.

(S94) The CPU 50a resets the Unopened flag of the message to the OFF state.

Through the above process steps, any attempt to delete an e-mail messages with an unopened must-see file attachment causes a warning dialog box, as well as opening and showing the attachment. This prevents the recipient 5 from deleting important files accidentally.

The next section will now focus on the primitive processes of the plug-in "TickerPlugin" shown in FIG. 6. The receiving station loads and executes this plug-in when it receives an e-mail message with a must-read e-mail 10 message. When the sender particularly wishes to call the recipient's attention, he/she inserts a composite keyword "<TICKER>" to the message's subject line to make sure that the recipient opens the message and see its file attachment.

15 Suppose first that the receiving station 50 has received an e-mail message with a keyword "<TICKER>." FIG. 23 is a flowchart showing a process to be executed in such a situation, which comprises the following steps.

(S100) The CPU 50a extracts the title of the received e-mail message. One example of this is as follows.

20 -- <TICKER> \*\* New PC Shop Open! \*\* East Computer Mart --

(S101) The CPU 50a removes the keyword from the title string obtained at step S100. In the present example, 25 the keyword "<TICKER>" is removed from the above title string, thus yielding the following text.

-- \*\* New PC Shop Open! \*\* East Computer Mart --

(S102) The CPU 50a opens a ticker window on the monitor unit 55. In the present example, it opens a new window which is large enough to accommodate the title text (i.e., without a keyword).

5 (S103) The CPU 50a displays the title text using what is called "ticker taping" techniques. FIG. 24 shows a typical ticker window 140 displayed on the monitor unit 55 when the above steps are executed. This ticker window 140 is titled "TICKER" and has a  
10 "Close" button 140a in the upper right corner for closing the window 140 itself. The pane 140b shows the following title text, which was obtained at step S101.

-- \*\* New PC Shop Open! \*\* East Computer Mart --

15 Within the pane 140b, this text is displayed in "ticker" mode; i.e., it runs from right to left across the pane 140b.

Through the above process steps, the receiving station 50 produces a ticker window attracting the  
20 recipient's attention. This means that the sender can differentiate his/her e-mail message from others by simply adding a keyword "<TICKER>" to it. While the above example is a plain character string, the keyword may include some arguments that represent the character sizes, colors, and  
25 other properties. This expanded keyword description will permit the ticker to gain more expressive power.

The above-described preferred embodiment of the

present invention has illustrated a system where the sender adds a special keyword as one of the message attributes (e.g., title), and at the receiving end, some primitive processes corresponding to the keyword are  
5 loaded and executed in response to relevant events. This mechanism provides the recipient with various services in accordance with the intention of the message. Furthermore, according to the present invention, the distributing station centrally manages primitive process description  
10 and keyword lookup table and distributes them to each receiving station. It is therefore possible to unify the behavior of receiving stations, and in addition, the definitions of primitive processes can be updated all at once, if so required.

15 While the above embodiment has illustrated an e-mail processing application, it is not intended to limit the present invention to this specific type of applications. Also, the location of keywords is not to be restricted to the title of a message, but alternatively,  
20 they can be inserted to the body of a message or added as part of the header of a message.

Typically, the above-described message processing mechanisms are implemented on a computer platform. The functions of a receiving station are actually realized as  
25 software functions of a computer system. Every process step can be encoded in a computer program, and such a program will be stored in a computer-readable storage

medium. The computer system executes the program to provide the intended functions of the present invention.

The suitable computer-readable storage media include magnetic storage media and solid state memory devices. Some portable storage media, such as CD-ROMs (Compact Disk Read Only Memory) and floppy disks, are also suitable for circulation purposes. Further, it will be possible to distribute programs through an appropriate server computer deployed on a network. The program file delivered to a user is normally installed in his/her computer's hard drive or other local mass storage devices, which will be executed after being loaded to the main memory.

The above discussion will now be summarized as follows. The present invention proposes an improved message processing apparatus for receiving and displaying messages sent from a sending end. This message processing apparatus comprises a receiver, a key information extraction unit, a program loading unit, and an execution unit. The receiver receives a message sent from the sending end. The key information extraction unit extracts key information from each message received by the receiver. The key information is contained in the message as part of attribute data of the message, and used to identify a predetermined handler program relevant to the message. The program loading unit loads a predetermined handler program associated with the key information. The execution unit

executes the predetermined handler program loaded by the loading unit at prescribed times. Configured as such, the proposed apparatus provides the recipient with appropriate processing services depending on the content of each  
5 message.

The foregoing is considered as illustrative only of the principles of the present invention. Further, since numerous modifications and changes will readily occur to those skilled in the art, it is not desired to limit the  
10 invention to the exact construction and applications shown and described, and accordingly, all suitable modifications and equivalents may be regarded as falling within the scope of the invention in the appended claims and their equivalents.

WHAT IS CLAIMED IS:

1. A message processing apparatus for receiving and displaying messages sent from a sending end,  
5 comprising:

reception means for receiving a message sent from the sending end;

key information extraction means for extracting key information from the message received by said 10 reception means, the key information being contained in the message as part of attribute data thereof and used to identify a predetermined handler program relevant to the message;

program loading means for loading the 15 predetermined handler program associated with the key information; and

execution means for executing the predetermined handler program loaded by said loading means at prescribed times.

20

2. The message processing apparatus according to claim 1, wherein:

the handler program provides at least one primitive process; and

25 said execution means selectively executes some of the primitive processes, depending on events that may occur subsequently to the reception of the message.

3. The message processing apparatus according to claim 1, wherein:

5 said program loading means has a table that defines relationships between the key information and the handler programs associated therewith; and

said program loading means consults said table to select and load one of the handler programs that is associated with the given key information.

10

4. The message processing apparatus according to claim 1, wherein:

the attribute data further includes a predetermined argument; and

15 said execution means executes the handler program, referring to the argument.

5. The message processing apparatus according to claim 1, wherein:

20 the handler programs are stored in a distributing station; and

the message processing apparatus further comprises downloading means for downloading the handler programs from the distribution station.

25

6. The message processing apparatus according to claim 5, wherein:

100-10528560

      said program loading means has a table that defines relationships between the key information and the handler programs associated therewith;

5      said program loading means consults the table to select and load one of the handler programs that is associated with the given key information; and

      said downloading means downloads the table, together with the handler programs.

10            7.      A computer-readable medium storing a computer program for receiving and displaying messages sent from a sending end, the computer program being designed to run on a computer in order to cause the computer to function as:

15            reception means for receiving a message sent from the sending end;

      key information extraction means for extracting key information from the message received by said reception means, the key information being contained in 20 the message as part of attribute data thereof and used to identify a predetermined handler program relevant to the message;

      program loading means for loading the predetermined handler program associated with the key 25 information; and

      execution means for executing the predetermined handler program loaded by said loading means at prescribed

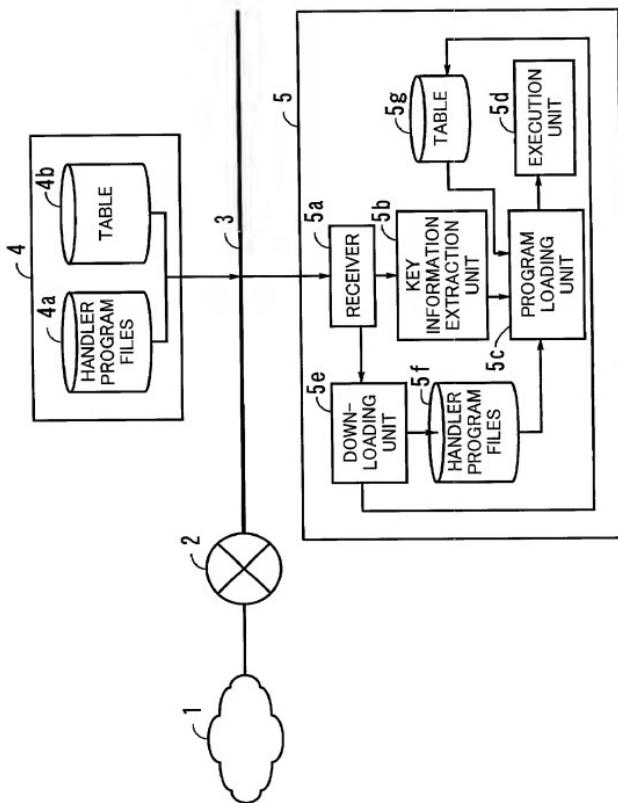
times.

UNIVERSITY LIBRARY

ABSTRACT OF THE DISCLOSURE

A message processing apparatus which handles messages exchanged over a network and explicitly notifies  
5 the recipient of the importance and other information regarding each received message. Incoming messages are received by a receiver employed in the apparatus. Those messages may contain some key information as part of their attribute data, which specifies a handler program relevant  
10 to them. A key information extraction unit extracts the key information from each received message. Consulting a table that defines the relationships between handler programs and various key information, a program loading unit then loads a specific handler program relevant to the  
15 extracted key information. The recipient may cause an event related to the received message. Depending on what kind of event it is, an execution unit chooses and executes an appropriate primitive process provided as part of the handler program. When, for example, the recipient  
20 attempts to delete an unopened message, the apparatus will warn him/her by displaying a warning message on a monitor screen.

FIG.1



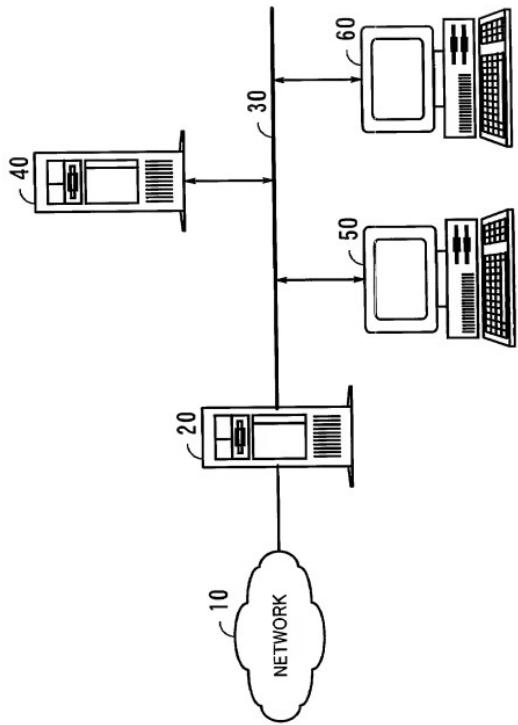


FIG.2

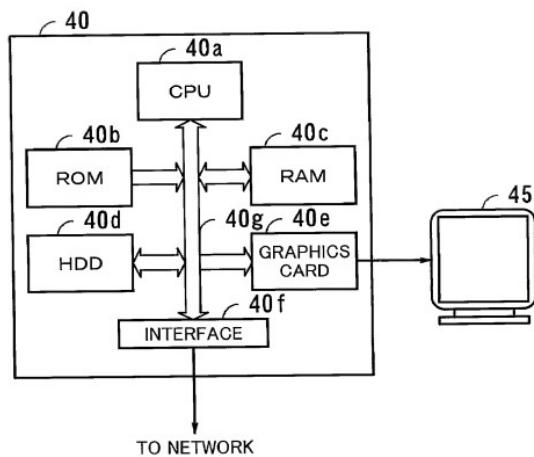


FIG.3

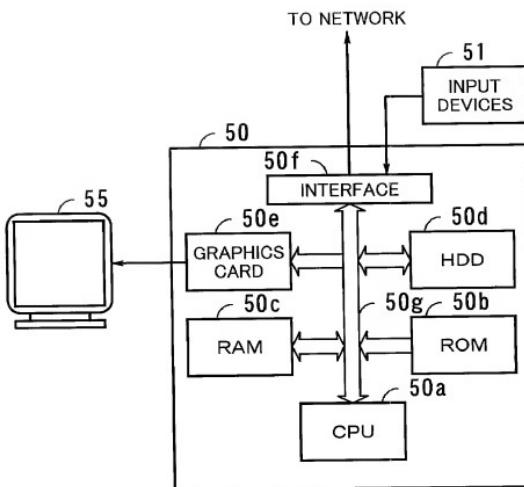


FIG.4

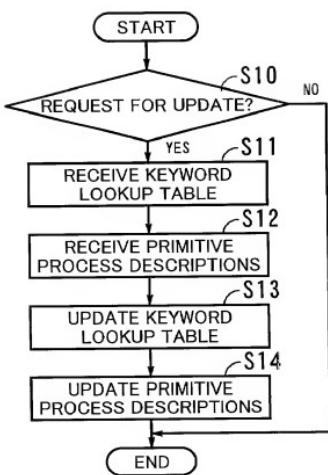


FIG.5

FIG.6

TYPE	PRIMARY SUB-KEYWORD	SECONDARY SUB-KEYWORD	ASSOCIATED PLUG-IN
MAIL	<URGENT:	BY ****>	TWPURgentPlugin
MAIL	<MUST-READ	>	MustReadPlugin
MAIL	<TICKER	>	TickerPlugin
:	:	:	:
:	:	:	:
FORUM	<MUST-READ	>	MustReadPlugin

```
Public TWPURgent Plugin implements TWPPPlugin{
    // INITIALIZATION
    public void init(TWPMessage Context context, String arg);
    // MESSAGE RECEPTION
    public void arrived();
    // APPLICATION RESTART
    public void restarted();
    // MESSAGE SELECTION
    public void selected();
    // MESSAGE DELETION
    public void deleted();
}
```

FIG.7

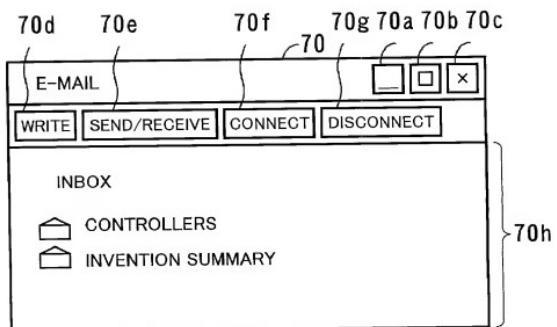


FIG.8

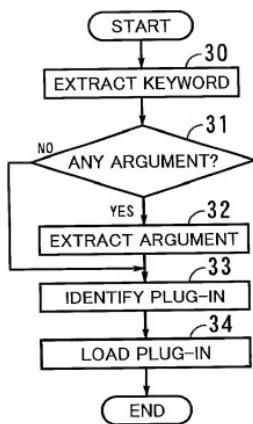


FIG.9

From:Manager<manager@bar>  
To:Foo<foo@bar>  
Date:Fri 11 Jul 1999 12:00:00 +900  
Subject:<URGENT: BY 99/06/14>Correction of Document

The invention summary I sent you the other day  
needs the following corrections...

.  
.

FIG.10

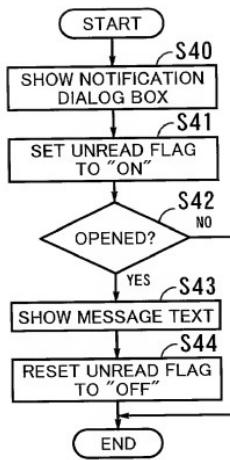


FIG.11

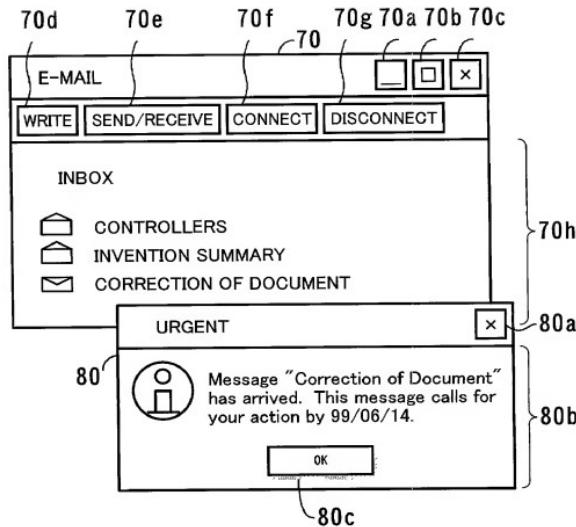


FIG.12

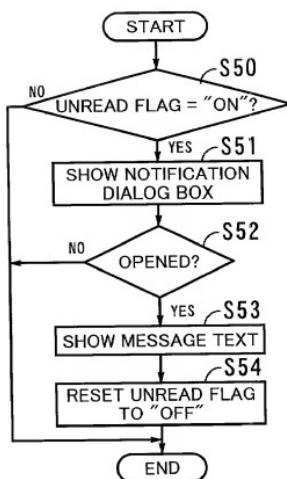


FIG.13

000760143522560

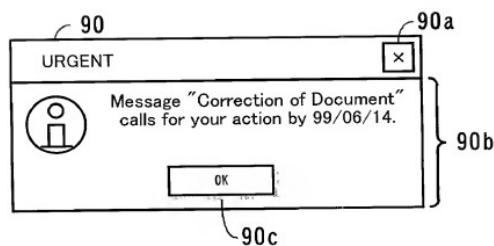


FIG.14

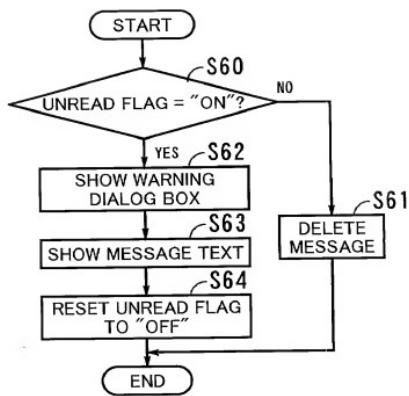


FIG.15

000000 - 000000 - 000000

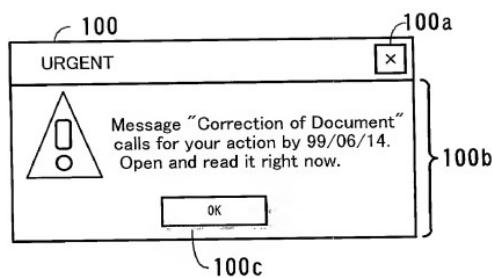


FIG.16

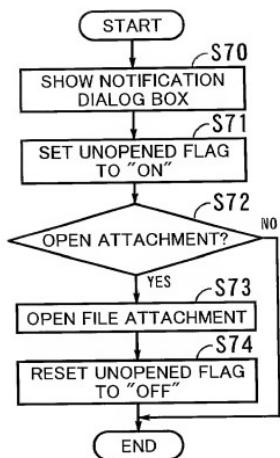


FIG.17

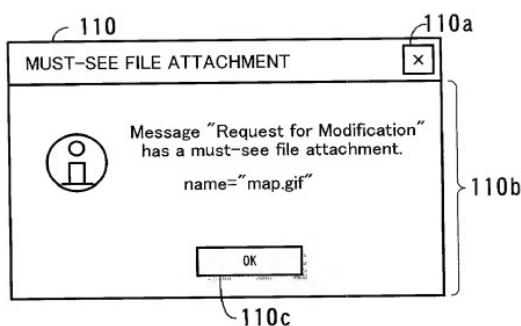


FIG.18

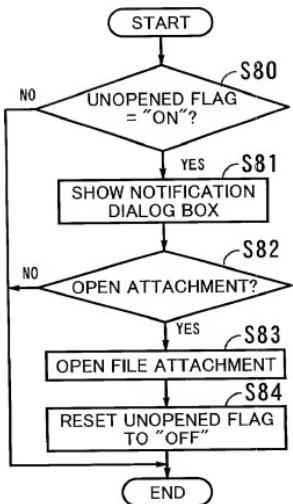


FIG.19

33001420 = 48522560

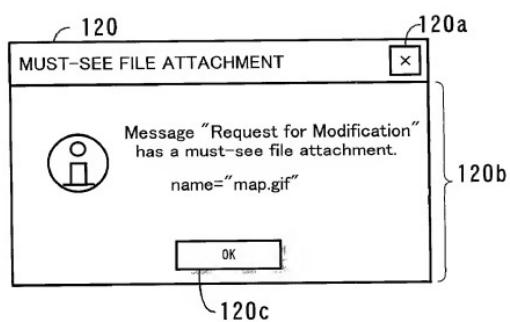


FIG.20

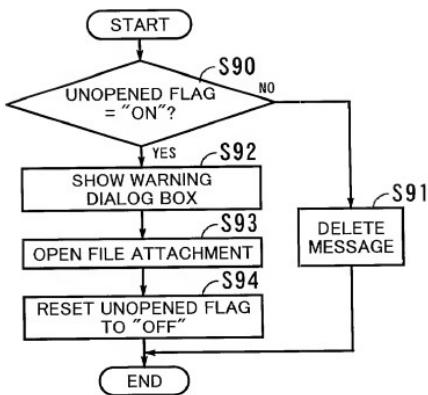


FIG.21

00000000000000000000000000000000

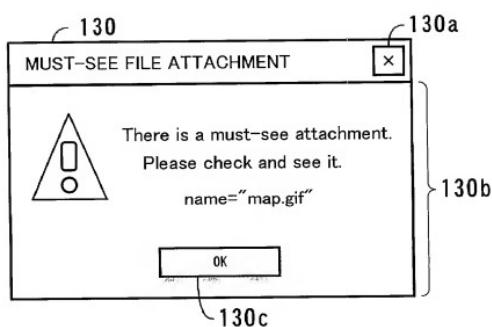


FIG.22

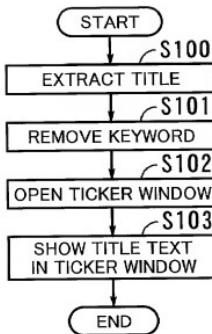


FIG.23

FIG 24



## Declaration and Power of Attorney For Patent Application

特許出願宣言書及び委任状

## Japanese Language Declaration

## 日本語宣言書

下記の氏名の発明者として、私は以下の通り宣言します。

As a below named inventor, I hereby declare that:

私の住所、郵便番号、国籍は下記の私の氏名の後に記載された通りです。

My residence, post office address and citizenship are as stated next to my name.

下記の名前が発明者として請求範囲に記載され、特許出願している発明内容について、私が最初かつ唯一の発明者（下記の氏名が一つの場合）もしくは最初かつ共同発明者であると（下記の名称が複数の場合）信じています。

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint Inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

## MESSAGE PROCESSING APPARATUS

上記発明の明細書（下記の欄でx印がついていない場合は、本書に添付）は、

the specification of which is attached hereto unless the following box is checked:

 \_\_\_\_月\_\_\_\_日に提出され、米国出願番号または特許協定条約国際出願番号を\_\_\_\_\_とし、  
(該当する場合) \_\_\_\_\_に訂正されました。 was filed on \_\_\_\_\_  
as United States Application Number or  
PCT International Application Number  
and was amended on  
\_\_\_\_\_ (If applicable).

私は、特許請求範囲を含む上記訂正後の明細書を検討し、内容を理解していることをここに表明します。

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

私は、連邦規則法典第37編第1条56項に定義されるおり、特許実格の有無について重要な情報を開示する義務があることを認めます。

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

## Japanese Language Declaration

## (日本語宣言書)

私は、米国法典第35編119条(a)-(d)項又は365条(b)項に基いて下記の、米国以外の國の少なくとも一ヶ国を指定している特許協力条約365(a)項に基づく国際出願、又は本国での特許出願もしくは発明者証の出願についての外国優先権をここに主張するとともに、優先権を主張している、本出願の前に出願された特許もまたは発明者証の外国出願を以下に、枠内をマークすることで、示しています。

## Prior Foreign Application(s)

外国での先行出願

11-207308

Japan

(Country)

(国名)

(Number)

(番号)

(Country)

(国名)

22/07/1999

(Day/Month/Year Filed)

(出願年月日)

Priority Not Claimed

優先権主張なし

(Day/Month/Year Filed)

(出願年月日)

私は、第35編米国法典119条(e)項に基いて下記の米国特許出願規定に記載された権利をここに主張いたします。

(Application No.)  
(出願番号)(Filing Date)  
(出願日)(Application No.)  
(出願番号)(Filing Date)  
(出願日)

私は、下記の米国法典第35編120条に基いて下記の米国特許出願に記載された権利、又は米国を指定している特許協力条約365条(c)に基いて権利をここに主張します。また、本出願の各請求範囲の内容が米国法典第35編112条第1項又は特許協力条約で規定された方法で先行する米国特許出願に開示されていない限り、その先行米国出願書提出日並びに本出願書の日本国内または特許協力条約国提出日までの期間中に入手された、進歩規則法第37編1条56項で定義された特許実務の有無に関する重要な情報について開示義務があることを認識しています。

(Application No.)  
(出願番号)(Filing Date)  
(出願日)(Status: Patented, Pending, Abandoned)  
(現況: 特許可済、係属中、放棄済)(Application No.)  
(出願番号)(Filing Date)  
(出願日)(Status: Patented, Pending, Abandoned)  
(現況: 特許可済、係属中、放棄済)

私は、私自身の知識に基いて本宣言書中で私が行なう表明が真実であり、かつ私の入手した情報と私の信じるところに基づく表明が全て真実であると信じていること、さらに故意になされた虚偽の表明及びそれと同等の行為は米国法典第18編第1001条に基づき、罰金または拘禁、もしくはその両方により処罰されること、そしてそのような故意による虚偽の声明を行なえば、出願した、又は既に許可された特許の有効性が失われるることを認識し、よってここに上記のごとく宣誓を致します。

I hereby claim foreign priority under Title 35, United States Code, Section 119 (a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 366(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

Priority Not Claimed

優先権主張なし

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below.

(Application No.)  
(出願番号)(Filing Date)  
(出願日)

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s), or 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States, or PCT International application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of application.

(Status: Patented, Pending, Abandoned)  
(現況: 特許可済、係属中、放棄済)(Status: Patented, Pending, Abandoned)  
(現況: 特許可済、係属中、放棄済)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**Japanese Language Declaration**  
(日本語宣言書)

委任状： 私は下記の発明者として、本出願に関する一切の手続を米特許商標局に対して遂行する弁理士または代理人として、下記の者を指名いたします。（弁護士、または代理人の氏名及び登録番号を明記のこと）

**POWER OF ATTORNEY:** As a named Inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith (list name and registration number)

<b>Attorney</b>	<b>Reg. No.</b>
Patrick G. Burns	29,367
Roger Greer	26,174
Lawrence J. Crain	31,497
Steven P. Fallon	35,132

<b>Attorney</b>	<b>Reg. No.</b>
James K. Folker	37,538
Jonathan D. Feuchtwang	41,017
B. Joe Kim	41,895
Joel H. Bootzin	42,343

直接電話連絡先：（名前及び電話番号）

Send Correspondence to:  
Direct Telephone Calls to: (name and telephone number)

Patrick G. Burns, Esq.  
Greer, Burns & Crain, Ltd.  
Sears Tower - Suite 8660, 233 S. Wacker Dr.  
Chicago, IL 60606 (312) 993-0080

**唯一または第一発明者名**Full name of sole or first inventor  
Toyoaki FURUSAWA

発明者の署名	日付	Inventor's signature	Date
		Toyoaki Furusawa	Feb. 25, 2000
住所		Residence	
		Kawasaki-shi, Kanagawa, Japan	
国籍		Citizenship	
		Japanese	
収書籍		Post Office Address	
		c/o FUJITSU LIMITED	
		1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8588 Japan	

**第二共同発明者**Full name of second joint Inventor, Mary  
Haruhito NAGATA

第二共同発明者	日付	Second Inventor's signature	Date
		Haruhito NAGATA	Feb. 25, 2000
住所		Residence	
		Kawasaki-shi, Kanagawa, Japan	
国籍		Citizenship	
		Japanese	
収書籍		Post Office Address	
		c/o FUJITSU LIMITED	
		1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8588 Japan	

(第三以降の共同発明者についても同様に記載し、署名をすること)

(Supply similar information and signature for third and subsequent joint inventors.)

第三共同発明者名		Full name of third joint inventor, if any Toshihiro NISHIMURA	
第三共同発明者の署名	日付	Third inventor's signature <i>Toshihiro Nishimura</i>	Date Feb.25, 2000
住所	Residence Kawasaki-shi, Kanagawa, Japan		
国籍	Citizenship Japanese		
私書箱	Post Office Address c/o FUJITSU LIMITED 1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8588 Japan		
第四共同発明者名		Full name of fourth joint inventor, if any	
第四共同発明者の署名	日付	Fourth inventor's signature	Date
住所	Residence		
国籍	Citizenship		
私書箱	Post Office Address		
第五共同発明者名		Full name of fifth joint inventor, if any	
第五共同発明者の署名	日付	Fifth inventor's signature	Date
住所	Residence		
国籍	Citizenship		
私書箱	Post Office Address		
第六共同発明者名		Full name of sixth joint inventor, if any	
第六共同発明者の署名	日付	Sixth inventor's signature	Date
住所	Residence		
国籍	Citizenship		
私書箱	Post Office Address		